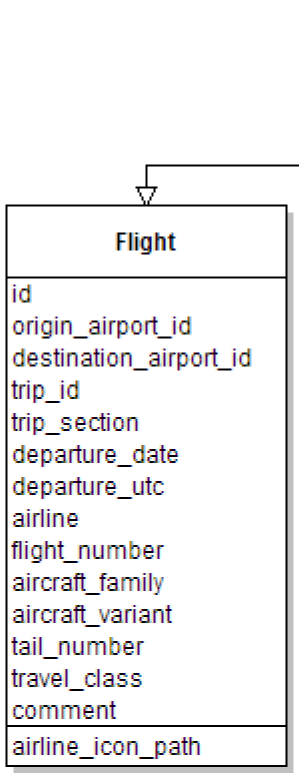
Flight Log Technical Specification

Paul Bogard · April 15, 2013

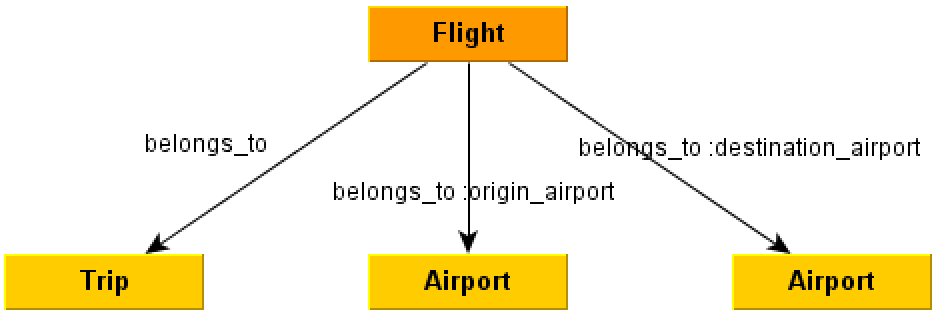
Contents

Classes



Flight

Associations



Attributes

Attribute	Type	Description
`id`	integer (required)	Unique flight identifier
`origin_airport_id`	integer (required)	Maps to the id attribute of

		Airport
`destination_airport_id`	integer (required)	Maps to the id attribute of Airport
`trip_id`	integer (required)	Maps to the id attribute of Trip
`trip_section`	integer (required)	Used to break a trip into subsections
`departure_date`	date (required)	Departure date of the flight (in the local time of the departure airport)
`departure_utc`	datetime (required)	UTC departure date and time, used to sort flights
`airline`	string	Airline operating the flight. For regional subsidiaries, use the parent airline; for codesharing, use the plane's livery.
`flight_number`	integer	The airline's assigned number for this flight
`aircraft_family`	string	Manufacturer and family type (e.g. "Boeing 737" and "Airbus A320")
`aircraft_variant`	string	Variant type and model (e.g. "737-800" and "A321")
`tail_number`	string	Tail number
`travel_class`	string	Class of travel (Economy, Business, or First)
`comment`	text	Comment

Methods

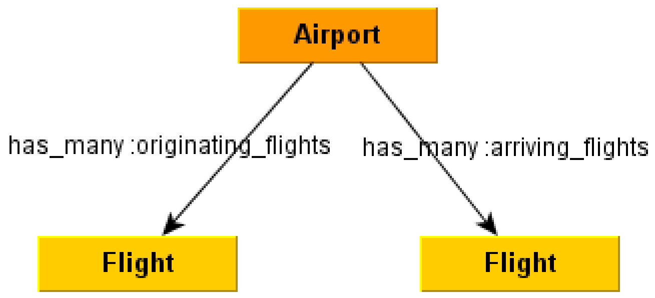
Standard Ruby on Rails ActiveRecord methods are available, but not listed in this document.

airline_icon_path()

Returns the path of this Flight's airline's logo icon as a string.

Airport

Associations



Attributes

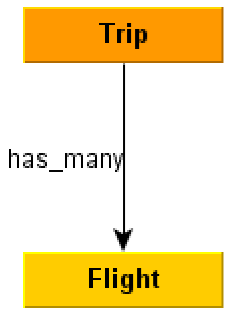
Attribute	Type	Description
`id`	integer (required)	Unique airport identifier
`iata_code`	string (required)	3-letter IATA code. Must be unique.
`city`	string (required)	Usually the city, with additional information if ambiguous (e.g. "Dayton" and "Chicago (O'Hare)" and "Portland, OR").
`region_conus`	bool	True if the airport is in the CONUS region, False otherwise

Methods

Standard Ruby on Rails ActiveRecord methods are available, but not listed in this document.

Trip

Associations



Attributes

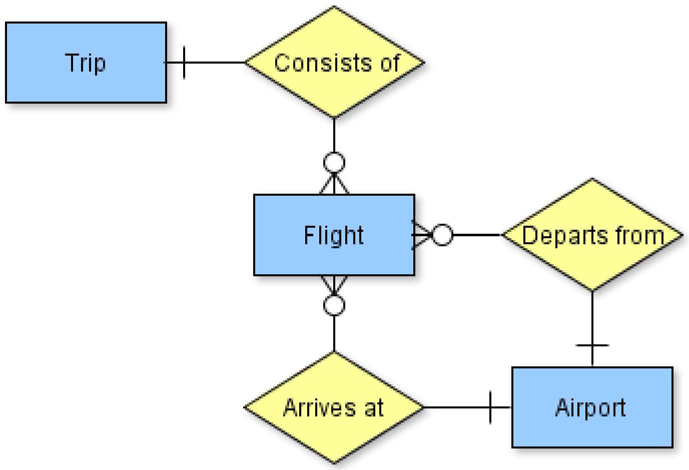
Attribute	Type	Description
`id`	integer (required)	Unique trip identifier
`name`	string (required)	Trip name
`hidden`	bool	True if the trip is only visible to verified users; False if visible to visitors
`comment`	text	Comment

Methods

Standard Ruby on Rails ActiveRecord methods are available, but not listed in this document.

Database

Entity Relationships



Size

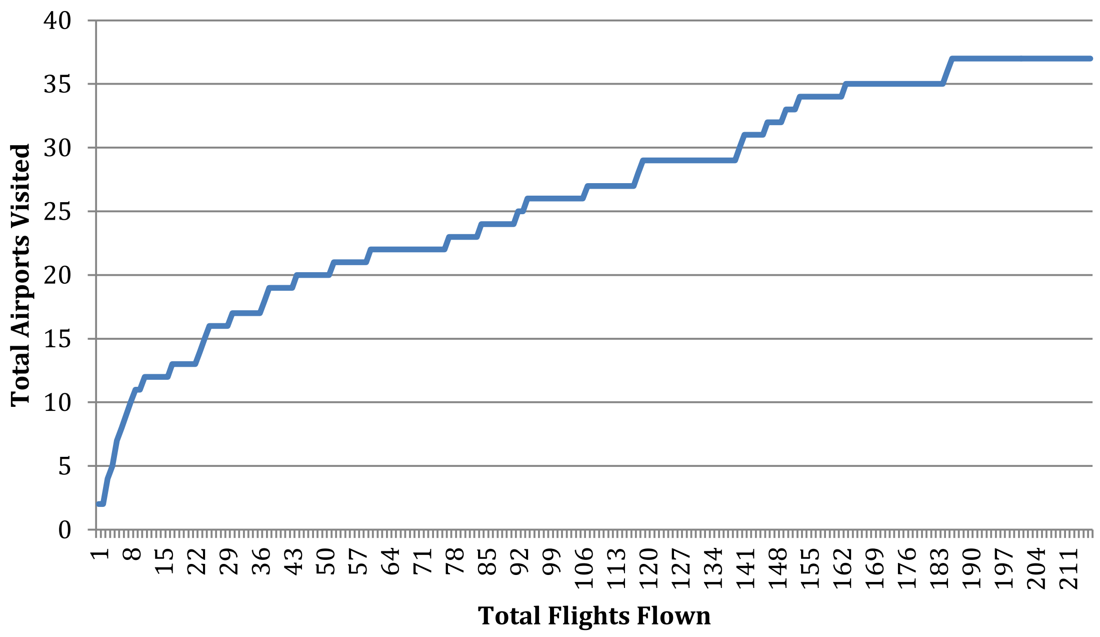
Per the functional specification, this site is intended for a single user (Paul Bogard), which will keep the size small.

For a conservative maximum number of records, assume an average of one trip per day and two flights per day for forty years.

$$\left(\frac{1\text{ trip}}{1\text{ day}}\right)\left(\frac{365.25\text{ days}}{1\text{ year}}\right)(40\text{ years}) = 14610\text{ trips}$$

$$\left(\frac{2\text{ flights}}{1\text{ day}}\right)\left(\frac{365.25\text{ days}}{1\text{ year}}\right)(40\text{ years}) = 29220\text{ flights}$$

At the time of the initial writing of this spec, Paul's flight log contained 219 flights and 37 airports. The number of airports as a function of flights appears to be less than linear. This is logical: the more flights are flown, the more likely it is that the flight will involve airports that have been visited in the past.



To get the worst-case prediction, though, we will assume a linear relationship with a ratio of 37 airports per 219 flights (and a y-intercept of zero).

$$29220 \text{ flights}\left(\frac{37 \text{ airports}}{219 \text{ flights}}\right) = 4937 \text{ airports}$$

Even at these extraordinarily worst-case numbers, these table sizes are easily within the capabilities of MySQL.